CyberDope: Optimizing Nucleotide Mixtures to Encode Specific Subsets of Amino Acids for Semi-Random Mutagenesis

Douglas C. Youvan

doug@youvan.com

November 18, 2023

CyberDope was originally written in C by Adam P. Arkin in my lab at MIT. A few years later, Mary M. Yang and I rewrote the program at Kairos Scientific Inc. to run in Windows. More recently, I rewrote the program from scratch in Mathematica. I release the Mathematica code here as CC4.0.

› Biotechnology (N Y). 1992 Mar;10(3):297-300. doi: 10.1038/nbt0392-297.

Optimizing nucleotide mixtures to encode specific subsets of amino acids for semi-random mutagenesis

A P Arkin [1], D C Youvan

Affiliations + expand

PMID: 1368102 DOI: 10.1038/nbt0392-297

Abstract

In random mutagenesis, synthesis of an NNN triplet (i.e. equiprobable A, C, G, and T at each of the three positions in the codon) could be considered an optimal nucleotide mixture because all 20 amino acids are encoded. NN(G,C) might be considered a slightly more intelligent "dope" because the entire set of amino acids is still encoded using only half as many codons. Using a general algorithm described herein, it is possible to formulate more complex doping schemes which encode specific subsets of the twenty amino acids, excluding others from the mix. Maximizing the equiprobability of amino acid residues contributing to such a subset is suggested as an optimal basis for performing semi-random mutagenesis. This is important for reducing the nucleotide complexity of combinatorial cassettes so that "sequence space" can be searched more efficiently. Computer programs have been developed to provide tables of optimized dopes compatible with automated DNA synthesizers.

```
(* Set partitioning on 64x12 genetic code and 3375x12 equimolar dope mixtures to separate 3 codon positions into groups of 4 variables for indexing *)


t=.; g=.;c=.;a=.;

gc=Tuples[{a,c,g,t},3];
a={0,0,0,1};
c={0,0,1,0};
g={0,1,0,0};
t={1,0,0,0};

(*
gc//MatrixForm; (* big; OK; stop codons at 57, 51, 49;  *)

gc[[57]];
gc[[51]];
gc[[49]];


{{1,0,0,0},{0,1,0,0},{0,0,0,1}}
{{1,0,0,0},{0,0,0,1},{0,1,0,0}}
{{1,0,0,0},{0,0,0,1},{0,0,0,1}}
*)

(* Make equimolar dopes, dp *)

tuptup=Tuples[Drop[Tuples[{0,1},4],1],3];

dope=Table[{0,0,0,0}, {i,1,3375},{j,1,3}];

For[i=1,i<=3375,i++, For[j=1,j<=3, j++,For[k=1,k<=4, k++,
   dope[[i,j]][[k]] = tuptup[[i,j]] [[k]]/ Total[tuptup[[i,j]][[All]] ] ] ] ];


dope=Partition[Flatten[dope],12];
gc=Partition[Flatten[gc],12];

(*
Dimensions[dope]
dope[[1]]
dope[[3375]]

{3375,12}
{0,0,0,1,0,0,0,1,0,0,0,1}
{1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4}
```

```
*)
(* Generate one matrix with all possible dopes and amino acid probabilities : 3375 dopes x (21 aa probabilities in alphabetic order with stop as ... w, x, y plus 12 columns of nucleotide mixture and a counter) = 3375 x 34 matrix rank 1 by partioning 3375 x 34 = 114,750 elements "ad" for All Dopes.  Matrix "dope" carries the first 12 columns of matrix "ad" and can be loaded now. Matrix "data" carries the next 21 columns and can be added to "ad" with iterations of a For statement, one row at a time.*)

ad=.;
ad = Partition[Table[{0}, {i,1,114750}],34];
(*
Dimensions[ad]
{3375,34,1}
*)


For[i=1, i<=3375, i++,
  ad[[i]][[1;;12]] = dope[[i]][[1;;12]] ];

(*
ad[[3375]]
Dimensions[ad]

{1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,1/4,{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0},{0}}
{3375,34}
*)
For[i=1,i<=3375, i++,

 dopegc=Table[{0,0,0,0,0,0,0,0,0,0,0,0}, {g,1,64}];
 For[g=1,g<=64,g++,dopegc[[g]]= dope[[i]]*gc[[g]]];
 pcodon=Table[{0},{g,1,64}];

For[g=1,g<=64,g++,pcodon[[g]]=(dopegc[[g]][[1]]+dopegc[[g]][[2]]+dopegc[[g]][[3]]+dopegc[[g]][[4]])* (dopegc[[g]][[5]]+dopegc[[g]][[6]]+dopegc[[g]][[7]]+dopegc[[g]][[8]])*
    (dopegc[[g]][[9]]+dopegc[[g]][[10]]+dopegc[[g]][[11]]+dopegc[[g]][[12]])];

 (* for i=3375
 pcodon


{1/64,1/64,1/64,
```

```
 data= { pcodon[[37]] + pcodon[[38]] +pcodon[[39]] + pcodon[[40]], pcodon[[58]] +
pcodon[[60]], pcodon[[34]] + pcodon[[36]], pcodon[[33]] + pcodon[[35]], pcodon[[62]] +
pcodon[[64]], pcodon[[41]] + pcodon[[42]] + pcodon[[43]] + pcodon[[44]], pcodon[[18]] +
pcodon[[20]], pcodon[[13]] + pcodon[[14]] + pcodon[[16]], pcodon[[1]] + pcodon[[3]],
pcodon[[29]] + pcodon[[30]] + pcodon[[31]] + pcodon[[32]] + pcodon[[61]] + pcodon[[63]],
pcodon[[15]], pcodon[[2]] + pcodon[[4]],pcodon[[21]] + pcodon[[22]] + pcodon[[23]] +
pcodon[[24]], pcodon[[17]] + pcodon[[19]],  pcodon[[9]] + pcodon[[11]] +pcodon[[25]] +
pcodon[[26]] + pcodon[[27]] + pcodon[[28]], pcodon[[10]] + pcodon[[12]] +pcodon[[53]] +
pcodon[[54]] + pcodon[[55]] + pcodon[[56]], pcodon[[5]] + pcodon[[6]] +pcodon[[7]] +
pcodon[[8]], pcodon[[45]] + pcodon[[46]] +pcodon[[47]] + pcodon[[48]],  pcodon[[59]],
pcodon[[49]] + pcodon[[51]] +pcodon[[57]], pcodon[[50]] +pcodon[[52]]} ;

 ad[[i]][[13;;33]] = data[[1;;21]] ; ad[[i]][[34]]=i;
 ]; (* end large i loop *)


(* For each of 3,375 rows of ad indexed by i, group probability (gp) is calculated and saved with
the index of i.  This forms a column vector 3375 x 1.  First, we form an equation for gp that is
based on whether each of the 21 aa's and stop are in the target set. The entire "ad" matrix is
copied to "adhold" so we can set the probabilities of the 21 aa plus stop variables to unity (1) if
they are outside the target set. This is sort of a surprising method for removing them from the
calculation. *)

adhold=ad;
gp =.;
gp=Table[{0},{i,1,3375}];
gpp=Table[{0},{i,1,3375}];


For[i=1, i <=3375, i++,c=0; For[j=13, j<=33, j++,If[ adhold[[i]][[j]] > 0, c=c+1,  adhold[[i]][[j]]
= -1] ];
 gp[[i]]= Abs[adhold[[i]][[13]] *  adhold[[i]][[14]] *  adhold[[i]][[15]] *  adhold[[i]][[16]] *
adhold[[i]][[17]] *  adhold[[i]][[18]] *  adhold[[i]][[19]] *  adhold[[i]][[20]] *  adhold[[i]][[21]]
*  adhold[[i]][[22]] *  adhold[[i]][[23]] * adhold[[i]][[24]] *  adhold[[i]][[25]] *
adhold[[i]][[26]] *  adhold[[i]][[27]] *  adhold[[i]][[28]] *  adhold[[i]][[29]] *  adhold[[i]][[30]]
*  adhold[[i]][[31]] *  adhold[[i]][[32]] *  adhold[[i]][[33]] ];  gpp[[i]] = gp[[i]] / ((1/c)^c) *100;
If[gpp[[i]]>=100 &&  c >= 12, Print[adhold[[i]][[1;;33]]," ",i," ", N[gp[[i]]]," ", N[gpp[[i]] ] ]
]  ]


(* gp[[3375]]  243/202824096036516704239472512860l6 *)
```

```
(* load default value of 1; note, 34 is not loaded *)

adhold[[All]][[13;;33]] = 1;

(* Search target set for aa's plus stop and load probabilities into adhold from matrix ad for all
3375 rows *)

a=.;c=.;d=.;e=.;f=.;g=.;h=.;i=.;k=.;l=.;m=.;n=.;p=.;q=.;r=.;s=.;t=.;v=.;w=.;x=.;y=.;

(* ////////////////////////////////////////////////////////////// *)
(*ts={a,c,d,e,f,g,h,i,k,l,m,n,p,q,r,s,t,v,w,x,y}; *)
ts={f,y,w};
(* ////////////////////////////////////////////////////////////// *)

tsd=Dimensions[ts][[1]];


For[ii=1, ii<=3375, ii++, For[jj=1,jj<=tsd, jj++,
   If[ts[[jj]] == a, adhold[[ii]][[13]] = ad[[ii]][[13]] ];
   If[ts[[jj]] == c, adhold[[ii]][[14]] = ad[[ii]][[14]] ];
   If[ts[[jj]] == d, adhold[[ii]][[15]] = ad[[ii]][[15]] ];
   If[ts[[jj]] == e, adhold[[ii]][[16]] = ad[[ii]][[16]] ];
   If[ts[[jj]] == f, adhold[[ii]][[17]] = ad[[ii]][[17]] ];
   If[ts[[jj]] == g, adhold[[ii]][[18]] = ad[[ii]][[18]] ];
   If[ts[[jj]] == h, adhold[[ii]][[19]] = ad[[ii]][[19]] ];
   If[ts[[jj]] == i, adhold[[ii]][[20]] = ad[[ii]][[20]] ];
   If[ts[[jj]] == k, adhold[[ii]][[21]] = ad[[ii]][[21]] ];
   If[ts[[jj]] == l, adhold[[ii]][[22]] = ad[[ii]][[22]] ];
   If[ts[[jj]] == m, adhold[[ii]][[23]] = ad[[ii]][[23]] ];
   If[ts[[jj]] == n, adhold[[ii]][[24]] = ad[[ii]][[24]] ];
   If[ts[[jj]] == p, adhold[[ii]][[25]] = ad[[ii]][[25]] ];
   If[ts[[jj]] == q, adhold[[ii]][[26]] = ad[[ii]][[26]] ];
   If[ts[[jj]] == r, adhold[[ii]][[27]] = ad[[ii]][[27]] ];
   If[ts[[jj]] == s, adhold[[ii]][[28]] = ad[[ii]][[28]] ];
   If[ts[[jj]] == t, adhold[[ii]][[29]] = ad[[ii]][[29]] ];
   If[ts[[jj]] == v, adhold[[ii]][[30]] = ad[[ii]][[30]] ];
   If[ts[[jj]] == w, adhold[[ii]][[31]] = ad[[ii]][[31]] ];
   If[ts[[jj]] == x, adhold[[ii]][[32]] = ad[[ii]][[32]] ];
   If[ts[[jj]] == y, adhold[[ii]][[33]] = ad[[ii]][[33]] ];
   ]];

(* Recalculate group probability with target set data and dummy 1's loaded *)

gp =.;
gp=Table[{0},{i,1,3375}];
For[i=1, i <=3375, i++,
```

```
  gp[[i]]= adhold[[i]][[13]] *  adhold[[i]][[14]] *  adhold[[i]][[15]] *  adhold[[i]][[16]] *
adhold[[i]][[17]] *  adhold[[i]][[18]] *  adhold[[i]][[19]] *  adhold[[i]][[20]] *  adhold[[i]][[21]]
*  adhold[[i]][[22]] *  adhold[[i]][[23]] * adhold[[i]][[24]] *  adhold[[i]][[25]] *
adhold[[i]][[26]] *  adhold[[i]][[27]] *  adhold[[i]][[28]] *  adhold[[i]][[29]] *  adhold[[i]][[30]]
*  adhold[[i]][[31]] *  adhold[[i]][[32]] *  adhold[[i]][[33]]
  ];

(* Combine adhold and gp matrices with 3375 gp values as column 35 in a new matrix, "sorted"
*)

sorted=Partition[Table[0,{ii,1, 118125}],35];
For[ii=1, ii<=3375, ii++, sorted[[ii]][[1;;34]] = adhold[[ii]][[1;;34]]; sorted[[ii]][[35]]=gp[[ii]] ];


(* Sort all rows in matrix "sorted" by highest value of the 35th column *)

sorthold=sorted;
For[ii=1, ii<=3375, ii++,
  For [jj=1, jj<=35, jj++,
   sorthold[[ii]][[jj]]=0]];

pad35=Table[0,{xx,1,35}];


For[ii=1, ii<=25, ii++,
  maxgp=-2;
  For [jj=1, jj<=3375, jj++, If[ sorted[[jj]][[35]] > maxgp, maxgp = sorted[[jj]][[35]] ; jjhi = jj;
    sorthold[[ii]] = sorted[[jjhi]] ] ; sorted[[jjhi]] = pad35 ]  ];




(* Matrix "output" contains only the top 25 dopes, otherwise we would have had to run a time-
consuming N x N sort to make matrix "sorthold".  At this point, we have sorthold loaded with
dummy vales of gp=1 so that aa's outside the target set do not enter into the specific gp
calculation, which is based only on probabilities for aa's within the target set.  Those values need
to be replaced. Also, we add a 36th column for % GP max based on 100% being an exact hit of
the dope on the target set, including equiprobable ratios of aa's as per Shannon's Entropy *)

output = Partition[Table[0, {ii,1, 900}],36];

For[ii=1, ii<=25, ii++, output[[ii]][[34]]= sorthold[[ii]][[34]]; output[[ii]][[35]]=
FortranForm[sorthold[[ii]][[35]]] ; output[[ii]][[1;;33]] = ad[[sorthold[[ii]][[34]] ]]  [[1;;33]];
output[[ii]][[36]] =  FortranForm[sorthold[[ii]][[35]]/((1/tsd)^tsd)* 100  ]   ];
```

```
TableForm[{N[Transpose[output[[1;;25]]],4]}, TableHeadings->{None,{"1T", "1G","1C", "1A",
"2T", "2G", "2C", "2A", "3T", "3G", "3C", "3A", "a",  "c",  "d",  "e",  "f",  "g",  "h",  "i",  "k",
"l",  "m",  "n",  "p",  "q",  "r",  "s",  "t",  "v",  "w",  "x", "y","#",  "GP",
"%GPmax"}},Dimensions[%36]]

(* Use Window Magnification to fit the entire table on your monitor. *)
```